SQL Project: Restaurant Order Analysis

Project Name

Restaurant Order Analysis-SQL

Objective

To analyze restaurant order data and understand customer preferences, sales trends, and popular menu items using SQL. The goal is to practice SQL operations such as table creation, joins, grouping, and aggregation on a small relational dataset.

Problem Statement

A restaurant wants to analyze its order data to make better business decisions. They want to know:
-Find Total Orders
-Show all Customers
- How much total revenue is earned
-What is average order value
- Find all customers from Mumbai
- List all items ordered in order ID = 101
-Which menu items are most popular
- Which customers are the most valuable
- How different cities are contributing to orders
-Sorting of customers alphabetically
-Category wise Revenue

Using SQL, we'll create the database, load the data, and answer these business questions.

Database Design/Tables Created and Inserted

1. customers-Table

```
CREATE TABLE customers (
  customer_id INT PRIMARY KEY,
  customer_name VARCHAR(100),
  gender CHAR(1),
  city VARCHAR(50)
);

INSERT INTO customers VALUES
(1, 'Rohan Mehta', 'M', 'Mumbai'),
(2, 'Priya Sharma', 'F', 'Delhi'),
(3, 'Amit Patel', 'M', 'Ahmedabad'),
(4, 'Neha Singh', 'F', 'Pune'),
(5, 'Arjun Nair', 'M', 'Bangalore');
```

Columns: customer_id, customer_name, gender, city
Sample Data:
1 | Rohan Mehta | M | Mumbai
2 | Priya Sharma | F | Delhi
3 | Amit Patel | M | Ahmedabad
4 | Neha Singh | F | Pune
5 | Arjun Nair | M | Bangalore

2. Menu_items-Table

```
CREATE TABLE menu_items (
  item_id INT PRIMARY KEY,
  item_name VARCHAR(100),
  category VARCHAR(50),
  price DECIMAL(10,2)
);

INSERT INTO menu_items VALUES
(1, 'Margherita Pizza', 'Pizza', 350),
(2, 'Farmhouse Pizza', 'Pizza', 450),
(3, 'Veg Burger', 'Burger', 150),
(4, 'Chicken Burger', 'Burger', 200),
(5, 'French Fries', 'Snacks', 100),
(6, 'Cold Coffee', 'Beverage', 120),
(7, 'Brownie', 'Dessert', 180);
```

Columns: item_id, item_name, category, price
Sample Data:
1 | Margherita Pizza | Pizza | 350
2 | Farmhouse Pizza | Pizza | 450
3 | Veg Burger | Burger | 150
4 | Chicken Burger | Burger | 200
5 | French Fries | Snacks | 100
6 | Cold Coffee | Beverage | 120
7 | Brownie | Dessert | 180

3 Orders-Table

```
CREATE TABLE orders (
  order_id INT PRIMARY KEY,
  customer_id INT,
  order_date DATE,
  total_amount DECIMAL(10,2),
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

```
INSERT INTO orders VALUES
(101, 1, '2024-01-10', 820),
(102, 2, '2024-01-11', 670),
(103, 3, '2024-01-12', 470),
(104, 1, '2024-01-13', 270),
(105, 4, '2024-01-13', 720),
(106, 5, '2024-01-14', 420);
```

Columns: order_id, customer_id, order_date, total_amount
Sample Data:
101 | 1 | 2024-01-10 | 820
102 | 2 | 2024-01-11 | 670
103 | 3 | 2024-01-12 | 470
104 | 1 | 2024-01-13 | 270
105 | 4 | 2024-01-13 | 720
106 | 5 | 2024-01-14 | 420

4 Order_details-Table

```
CREATE TABLE order_details (
  order_id INT,
  item_id INT,
  quantity INT,
  FOREIGN KEY (order_id) REFERENCES orders(order_id),
  FOREIGN KEY (item_id) REFERENCES menu_items(item_id)
);
```

```
INSERT INTO order_details VALUES
(101, 1, 1),
(101, 5, 2),
(101, 6, 1),
(102, 2, 1),
(102, 7, 1),
(103, 3, 2),
(104, 5, 1),
(104, 6, 1),
(105, 4, 2),
(105, 7, 1),
(106, 3, 1),
(106, 6, 2);
```

Columns: order_id, item_id, quantity
Sample Data:
101 | 1 | 1
101 | 5 | 2
101 | 6 | 1

102 | 2 | 1
102 | 7 | 1
103 | 3 | 2
104 | 5 | 1
104 | 6 | 1
105 | 4 | 2
105 | 7 | 1
106 | 3 | 1
106 | 6 | 2

Verify(Check the execution)

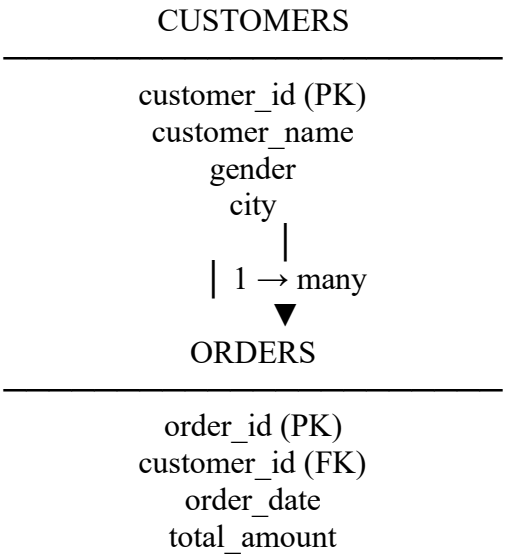
Try these quick tests:

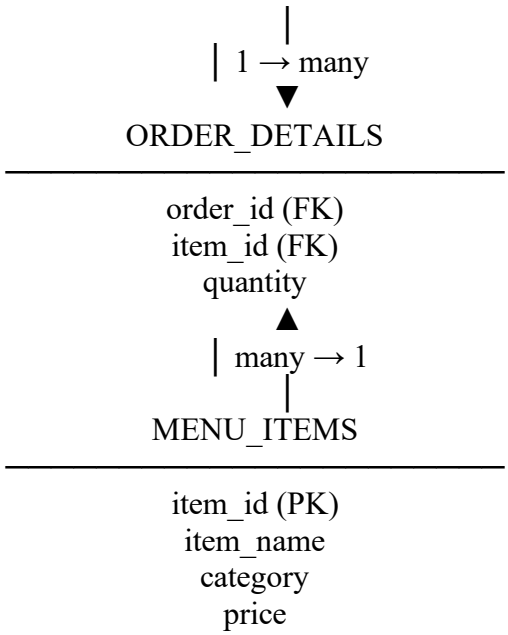
```
SELECT * FROM customers;
SELECT * FROM menu_items;
SELECT * FROM orders;
SELECT * FROM order_details;
```

Relationships / E-R Diagram

E.R. Diagram – Restaurant Order Analysis

The following Entity-Relationship Diagram represents the relational structure of the Restaurant Order Analysis database. It highlights key tables and their relationships (One-to-Many connections).





Relationship Summary

• CUSTOMERS(1) → ORDERS(Many) = One-to-Many (A customer can place many orders)
• ORDERS(1) → ORDER_DETAILS(Many) = One-to-Many (An order can contain many items)
• MENU_ITEMS(1) → ORDER_DETAILS(Many) = One-to-Many (A menu item can appear in many orders)

Relationships:
- customers (1) → (many) orders
- orders (1) → (many) order_details
- menu_items (1) → (many) order_details

Basic E-R Structure:

CUSTOMERS (customer_id)
|
| 1 --- many
|
ORDERS (order_id, customer_id)
|
| 1 --- many
|
ORDER_DETAILS (order_id, item_id)
|
| many --- 1

|
MENU_ITEMS (item_id)

Basic SQL Queries and Insights

1]-----Question: Find total orders

Query: SELECT COUNT (*) FROM orders;

Output:
total_orders-6

Insight: 6 orders placed

2]-----Question: Show all customers

Query: Select * from customers;

Output:
Customer id | Customer Name |Gender |City
'1','Rohan Mehta', 'M' ,'Mumbai'
'2','Priya Sharma', 'F' ,' Delhi'
'3','Amit Patel', 'M' ,'Ahmedabad'
'4','Neha Singh' ,'F' ,'Pune'
'5','Arjun Nair' ,'M', 'Bangalore'
Null, Null , Null ,Null

Insight- Knowledge about customer-base generated

3]-----Question: How much total revenue is earned?
Query: SELECT SUM (total_ amount) FROM orders;

Output:
total_ revenue
3370

Insight: ₹3370 is the total revenue

4]-----Question: Average order value/What is the average order value?
Query: SELECT AVG (total_ amount) FROM orders;
Output:
Avg(total_amount)=561.67

Insight: ₹561.67 average

5]-----Question: Find all customers from Mumbai?
Query: SELECT customer_name FROM customers
WHERE city = 'Mumbai';

Output: customer_name
Rohan Mehta

Insight: Rohan Mehta lives in Mumbai.

6]----- List all items ordered in order ID = 101
Query: SELECT m.item_name, od.quantity
FROM order_details od
JOIN menu_items m ON od.item_id = m.item_id
WHERE od.order_id = 101;

Output: item_name | Quantity
'Margherita Pizza','1'
'French Fries','2'
'Cold Coffee','1'

Insights-Margherita Pizza, French Fries and Cold Coffee are items ordered by order id 101

7]-----Question: Item popularity/ Which menu items are most popular?

Query: SELECT m.item_name, SUM(od.quantity) FROM order_details od JOIN menu_items m ON od.item_id = m.item_id GROUP BY m.item_name;

Output:
Item name|order quantity
'Margherita Pizza','1'
'Farmhouse Pizza','1'
'Veg Burger','3'
'Chicken Burger','2'
'French Fries','3'
'Cold Coffee','4'

'Brownie','2'

Insight: Fries & Coffee most ordered

8]-----Question: Which Customer spends the most?
Query: SELECT c.customer_name, SUM(o.total_amount) AS total_spent
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
GROUP BY c.customer_name
ORDER BY total_spent DESC
LIMIT 1;

Output:
Customer_ name | total_ spent
'Rohan Mehta','1090.00'

Insight: Rohan Mehta spent most

9]-----Question: Find Orders per city/How different cities are contributing to orders
Query: SELECT c.city, COUNT(o.order_id) FROM orders o JOIN customers c ON o.customer_id = c.customer_id GROUP BY c.city;

Output:
City| Count(o.order_id)
'Mumbai','2'
'Delhi','1'
'Ahmedabad','1'
'Pune','1'
'Bangalore','1'

Insight: Mumbai has highest orders(Group By used since question is orders per city)

10]-----Question: Sort Customers alphabetically
SELECT customer_name FROM customers
ORDER BY customer_name ASC;

Output:
Customer_name
'Amit Patel'
'Arjun Nair'
'Neha Singh'
'Priya Sharma'
'Rohan Mehta'

Insight- Customers are sorted alphabetically from Amit Patel to Rohan Mehta

-----Extra Question(Little Complicated) Category-wise revenue
Query:

```
SELECT m.category, SUM(m.price * od.quantity) AS category_revenue
FROM order_details od
JOIN menu_items m ON od.item_id = m.item_id
GROUP BY m.category
ORDER BY category_revenue DESC;
```

Output:
Category| Category_ revenue
'Burger','850.00'
'Pizza','800.00'
'Beverage','480.00'
'Dessert','360.00'
'Snacks','300.00'

Insight- Burger generates a lot of revenue

📊 Insights Generated

- Total 6 orders and ₹3370 revenue recorded.
- Pizza and Burger categories drive major sales.
- Rohan Mehta (Mumbai) is the top spender.
- French Fries and Cold Coffee are popular add-ons.
- Mumbai contributes the highest number of orders.

🏁 Conclusion

This project demonstrates how basic SQL operations can be used to analyze restaurant performance. By understanding customer and product data, businesses can:
- Identify top-selling products
- Reward loyal customers
- Improve menu and pricing strategies which may help the restaurant in the long run.
-Know your customer demographics and serve them better